Software World Tour - CoRecursive Podcast

40:11 Estimated 8440 Words EN Language

Note: This podcast is designed to be heard. If you are able, we strongly encourage you to listen to the audio, which includes emphasis that's not on the page

Intro

Adam: Hi. This is CoRecursive and I'm Adam Gordon Bell.

Do you know the song, The Safety Dance by Men Without Hats? It's like (singing).

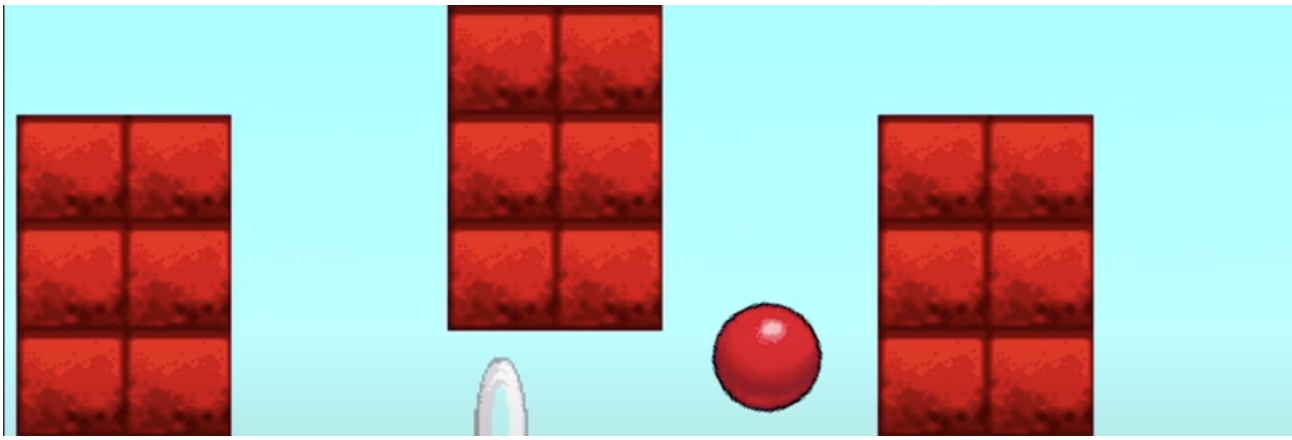
Early in my career, the second developer job that I had, it was in 2008. I worked at a web agency. It was a smallish office and they played music in the office over speakers, mainly, I think, to drown out the people on the phone, support people, salespeople. The owner picked the station and I think it was beamed right from 1982 because it was all '80s pop hits all the time.

I know this isn't true, but in my mind when I remember it, it was 50% or more The Safety Dance by Men Without Hats. I still have no idea what that song is about, but I left that job at some point and I went to Operitel. It was a software product company and the first day, I noticed how quiet it was. It was so nice.

The web agency was culturally like a sales place, but Operitel, it was quirky and it was nerdy. I could tell right away, I just fit in better there. But maybe I wouldn't have known it without The Safety Dance, without hearing that song so much.

That's what today's story is about. It's not a story of a piece of software being built, **it's a story of a software developer finding a place that fits them, a place that suits them**. It's also going to be a bit of a world tour where we spend some time in various countries, including Canada and China and, I don't want to spoil too much, but elsewhere.

Building Games



Bounce Game inspired Son

Adam: Can you tell me who you are and what you do?

Son: Yeah. My name is Son Luong. I contribute to various different open source project including Bazel. I have some contribution into Git and GitLab itself and yeah, that's a short summary.

Adam: The earliest glimmer of Son's career started with a lie. He was in high school at a friend's house and the friends showed him a game.

Son: I think the name of it was Bounce or something where you are a ball and you were just trying to get from one platform to another and eating candy or something like that, getting points.

Adam: His friend claimed that he knew how to make games and that he could make this game.

Son: He was completely BSing by the way. He cannot make that. You need a graphic team with images to build that game. He alone cannot build that.

Adam: So Son tried to make his own game.

Son: And I went home and I just keep remember like how the hell do I make anything work with this Visual Basic program, trying to drag the boxes and drag the button? This button is not the game that I'm looking to build. I can click it and create like a hello world. I can click it and create a text game where you answer yes or no and then choose your path, but that's not cool.

Night Classes

Adam: And so Son started taking some night classes in computer programming.

Son: Where they teach you how to make computer program like web server and stuff. I remember the teacher there was like, "Man, you're in grade 10. Why are you here?" My classmate were all final year university students and people who already have a job and trying to learn more. I was the only high school student there.

Adam: The class was fun, but making a game still seemed like a mystery. So when Son finished high school, because he was good at math and liked the idea of building a game, he went to the University of Waterloo in Waterloo, Ontario in Canada as a foreign student. Waterloo was a culture shock.

University of Waterloo

Son attended University of Waterloo's Mathematics Department

Son: Sitting next to me is one Asian guy who was born and raised in Canada and he was using Vim and LaTeX to take notes. I was like, "What the hell is LaTeX? What the hell is Vim?" And sitting on the left-hand side of me is an Indian guy who's taking a double degree program in both science and pure math. The guy end up being a doctor. I remember a conversation on first year. He was like, "Oh, did you buy this stock?" and I was like, "What? You're just turning 18, dude. How the hell do you know anything about stock?" Turns out his parent taught him all this stuff. I remember on third year, I met a couple who, after the final exam, was going to the library to borrow math books to go back and study after the final exam.

What the hell are you studying for? I was like, man, that entire environment was way too competitive. Coming from Asia, I was not expecting people to be that active. Learning in Asia was something very passive. Whatever you are given to by the teacher, you learn that and that's it, but in North America and I think in Europe, if you were to learn something, then you must have a passion for it because you have a choice of choosing to learn that topic. So once you have a passion for it, it means that you enjoy learning about it and you would spend a lot of time self-studying about it. I did not have any of that. Even though I love math, I didn't love it that much. I just love it simply because I was good at it.

Adam: This was Son's first lifestyle adjustment and there's going to be more, but finding his place at Waterloo was a challenge. It's a very serious school, but Son found a way to make it work.

Son: I remember I was having a really interesting conversation about, hey, what do you think gravity is? And people was actually replying to you like, "Okay, here are the theoretical physics about gravity."

The university taught me two things. One is obviously, the computer science stuff is quite cool, but what it taught me is I need to value this environment. When I enter a competitive environment, I should not be

competitive, but I should befriend all these people because turns out later in life when I look back, I cannot find the same quality of people, the same smartness, the same speed that I was able to find in Waterloo.

Adam: There's a path from being a foreign student in Canada to becoming a permanent resident and that was Son's goal, but all his friends who are Canadians seemed to be getting jobs in the US at Google or at Facebook or wherever. In Canada, we'd call this the brain drain and maybe with remote work, this'll slow down a little bit, but for Son, with his friends gone, he felt trapped in Waterloo and in Canada.

Son: So I was getting really depressed. I was starting straight for eight years without seeing my family at all, so no video calls, no air flight, nothing like that. So I decided, okay, I need a mental reset and this mental reset mean that I should go back to Vietnam and try to find something.

Back To Vietnam

Son travelled back to Hanoi, Vietnam

Adam: After the quiet life of a math department and a small city in Canada, being back in Vietnam seemed like the future.

Son: A lot of expats, French, American, Russian, everybody was moving to Vietnam or another Southeast Asian country because they like life over there. The weather is wonderful, beaches. If you ever had a chance to take a vacation, I definitely strongly recommend Southeast Asia.

Adam: During this time, at least in his head, Son was still keeping his game dev dreams alive.

Son: I begin in Vietnam starting to think like, hey, after one or two years of working there, I can collect enough money to open up a game company.

Adam: To build his bank roll, he started a job as a business analyst at CSE.

Son: They were making insurance type software for various different companies and turns out back in 2016, hello, insurance tech company was still running on COBOL. Turning that into Java is a hugely profitable and yeah, we were having deal with various different insurance company.

Adam: At the insurance company, Son started to see a problem with this game-making plan.

Son: When you get into insurance, you start learning the value of reliability, the value of being able to assure something. People pay serious money for that and I was like, why do all these people who are obviously very smart able to afford a lot of money to buy these insurance packages? Why do they value this so much?

Adam: As a Waterloo math grad, Son easily had the skills for understanding insurance and actuarial math, and it all started to make sense to him.

Son: There's some value here. You don't need to gamble everything, but there are percentage and there are value in safety net. You can actually quantitatively calculate that using an actuarial model behind all that insurance and then all that math from Waterloo start kicking it. I was like, "Oh, this all makes sense now. Okay, I can actually model out my life and my income and all that in a much better way than just gambling everything away."

Adam: And through this new lens, Son's plan to start a game dev company, it just didn't seem to make sense.

Son: I was like, "Oh, shit, that is actually gambling and there's a lot of risk into all that and I should not be pursuing that if I want to have a sustainable life, a healthy life."

Adam: So Son starts learning more software development, but he has a different perspective now because it's not about gaming, it's just about building things and learning. Maybe being a developer can be his calling if he just throws himself into it.

Lazada 911

Son started as an SRE at Lazada

Son: I had a lot of time to study and I was learning AWS, all that good stuff and I landed a job in Lazada, onto a team called 911. You can guess what that team do, emergency support for the tech platform. So

Lazada is an e-commerce company in Southeast Asia. It's either number one or number two at the time. I think they were number one back in 2017 and they were selling in six different countries.

Adam: Population wise, Southeast Asia is huge and diverse. It includes Vietnam and Thailand and Laos, but also Singapore and the Philippines, Malaysia, Indonesia and more. It's got 700,000 people and each of those countries, people speak a different language or have different customs. Lazada was serving that market out of their main office in Vietnam. A few hundred people worked there.

Son: I think at least 200 technical and then we have the business side as well. I remember we had at least three floor on a building and I think each floor was carrying around hundred people.

You get free flow of drinks. You have a cappuccino machine, but you don't need to work long hours unless you don't want it to. Plus it's like to work in Vietnam, very competitive, very cutthroat, competitive and I think it's not solely for Vietnam, but it's just like that for Asian countries in general.

The tech stack is a PHP Monolith was being converted to microservice Golang running on Kubernetes. Back in 2017, this is very modern, moving really fast.

We have an R&D center in Moscow. We have an R&D center in Vietnam and a data engineering center I think either in Thailand or Singapore or both. We have three different data center that we manage by ourself.

Adam: The 911 team, well, it had a clear job.

Son: Receiving escalation from business side, because we have business entity in six different countries. Including Hong Kong, it's seven.

We were receiving escalation from business entity and then we escalate that over to tech side and the job there was like to speed up the troubleshooting. So instead of 911 having to forward the ticket to the development team, then 911, just fix it. Just go there, find what's wrong, fix it immediately. I had my hands on all over the system and yeah, I was learning to be what now they would call an SRE back before Google released the definition of SRE.

But yeah, with the background in insurance, I was immediately able to recognize why business was valuing this, because they want that stability, they want that reliability and yeah, it is just a very interesting way for my previous job to somehow met the criteria of my new job and it was very fun. It was a very fun time.

Double Eleven Day

11/11 day (Singles Day) was huge at Lazada, especially for the 911 (SRE) team

Adam: For an e-commerce chain in Southeast Asia, the biggest day of the year isn't Black Friday. It's something else.

Son: Oh, yeah. Back in Southeast Asia, folks in North America and the European country don't know this, but Southeast Asia is heavily influenced by China and China have a big sales on Double 11 Day. So on the 11th of November, it's the biggest sales in China. It's similar to Black Friday in North America or Boxing Day in Canada. So Double 11th is also the biggest sales in Lazada at the time. We open up sales at midnight for all the countries and before that sales coming in, we had to do stress testing for month before that in able to prepare for the load.

The Problem

Adam: But on the big day, a problem occurred.

Son: The midnight of the 10th of November to the morning of the 11th of November, we detect the problem 30 or 20 minutes in. We were like, "Oh, we had a lot of order in Malaysia. Yay! Wait. Why everything stuck in pending payment? Wait a minute. Something is wrong here. Go back and check." Oh,

turns out everything was being invalidated by payment processor. Turns out on the night, our credit card processor, a third party credit card processor in Malaysia just went down. They just completely given up. All the transaction we had sent over there was completely disappearing. When you build up an e-commerce platform, you cannot build everything. You cannot build the e-logistics side. You cannot build the payment. You cannot build the notification part, so you need to use third party. When third party integration goes wrong, you need to have a backup plan for it.

At the time, we don't have one for Malaysia. We had a war room going on in the 911 room and everybody was there, the payment team, the CEO, the CTO. Everybody was in the room and I had to scan the logs to see what's wrong. The CEO and CTO was looking at me like, "Hey, look at the lock and see what's wrong," and turns out it was payment. All these was way before event-driven microservice. We had to repair text lock or JSON lock into some sort of pseudo CSV file and then we have to manually data fix the database and then replay all that state. Turns out if you turned off all the cron job that invalidate the order that was not paid for one hours, every other cron job also die in the system.

So yeah, the system back then was not at the best state, but that's how we keep all the orders. The revenue was huge and yeah, it was quite fun. It was quite a night. In the morning, we end up drinking a lot of Red Bull and then everybody was sleeping all over the place. We see the CEO was sleeping over there. The CTO was sleeping over here. Yeah, it was quite a night. I think it was 2017, I think. The 2018 has got a completely different story.

Adam: 2018 was different because Alibaba got involved. Alibaba had bought Lazada in 2016, but it had mainly left it alone. Southeast Asian software companies were used to this: to being left to do things their own way.

Son: When you're working in Southeast Asia, you need to move fast, but at the same time in Southeast Asia, you have six different countries speaking six different language with six different culture. How the hell do you move fast? That's very difficult. Instead of building one website, now you have to build six different website with eight different languages and at the same time, you have to deal with regulation of six different countries. For example, Indonesia, they have laws where similar to GDPR where the data cannot leave the border of the country, the physical border of the country. So now you have to go into the country and build a data center, and this is back before AWS was a thing over there. So you have to go in there. You need to establish a data center and then you need to deploy a different stack, an isolated stack compared to the rest of the country. Then now you have to handle data synchronization between different tenants in your system, different data center. It makes the problem a lot more complex and overcoming this complexity get you ahead.

Adam: It's reasons like that that Uber and many other hypergrowth companies lost money going into Southeast Asia. It's not just American companies who are out looking for growth. China has tech giants as well and Alibaba is a big one.

Enter Alibaba

Alibaba bought Lazada in 2016, leading to a replatform onto Java Spring

Son: Alibaba is the owner of Tmall and Taobao. That's the equivalent of Amazon. One is C2C and one is B2C. They own Ali Cloud. If you want to do business in China, you need to host that somewhere and that somewhere is Ali Cloud. Amazon footprint over there is very small. The comparison of Baidu and Tencent are very small. The most reliable one is Ali Cloud. Most of the major banks are tenants of Ali Cloud. Most of the major banks out there are hosting their services on Ali Cloud, so they were very big. Aside from that, they do various different business. For example, Youku is a video streaming platform that's the size of YouTube. They also operate something that most people don't know about called Cainiao. Cainiao is a logistic platform that handle all the shipping so stuff that you are buying on AliExpress today or even on the Amazon today most likely, that will ship over from China to your country to your port with Cainiao. That's a global logistic platform that they acquire. That's very big. They were all over the place. They were really big. They are a big tech giant.

Adam: The acquisition caused some friction at Lazada.

Son: The company made the decision sometime around 2016 and that's very early to move to use Go and Kubernetes instead of deploying on bare metal VM. So everything was very modernized and high volume, but was not that high compared to the like of Google or Amazon, but definitely, it was one of the highest [traffic] services at the time in Southeast Asia.

There were really a lot of change and change come with uncertainty and that's innovation. The rate of innovation was really high, so the platform in turn becomes a little bit less reliable. That's just a business risk that the entire group willing to accept. But once Alibaba came over, these folks are highly experienced. They are used to handling 100 time, 1,000 time higher traffic without sacrificing that resiliency. So they decide, "Hey, we want the system to be more reliable. Let's re-platform everything to something that we know that going to work."

Ali Java

Adam: For some developers, switching tech stacks was a big ask. It's throwing away a lot of knowledge, and Alibaba stack is all Java and Spring.

Son: Everybody was like, "Java sucks. We like Golang. Get the hell out of here. Where's your Kubernetes? Oh, you don't have Kubernetes? Where's your Docker? Oh, you have your own Docker? Nah." But it turns out, there are good stuff in Chinese tech that worth learning about and I think some of that, Westerners still don't know about today. It was quite an eyeopener.

Adam: An acquired company being forced to take their working software and re-platform it to however the parent company does things, it's a classic failure mode of acquisitions. The re-platform will take years and you'll end up with worse solutions because of assumptions built deep into how the acquired company did things. But in a bigger way, re-platforms just fail because they end up being bigger investments than the company thought they were going into it. And here's the thing about Alibaba. Alibaba was not afraid to invest in Lazada.

Son: That's just something that the company decided and they said target and they actually put quite significant investment into it. Later on, I learned that this project is actually the biggest project in Alibaba at the time. What they call it at the time was a C-level project and a C-level project, meaning that the CEO of Alibaba was actually quite invested into it and they managed to extract all the talents from all different business unit in Alibaba into a special team to make this happen. Other time I couldn't understand why the hell would you invest in Southeast Asia? But later on when I start working for Alibaba, I start realizing, oh, this is actually a much, much bigger picture than I ever imagined. Oh, my God. Turns out it related to a long-term planning that pivoting with the Chinese Belt and Road Initiative and yeah, it was way bigger than we ever imagined. It all makes sense after we learned about that, but before that, everybody was like, Java sucks.

Hangzhou, China

Son travelled to Alibaba's Xixi campus map - Image Source: Son

Adam: Everyone at Lazada starts learning about Alibaba-flavored Spring. For Son, this was a chance to learn about a whole different world of software development, how things worked at the scale of Alibaba Group.

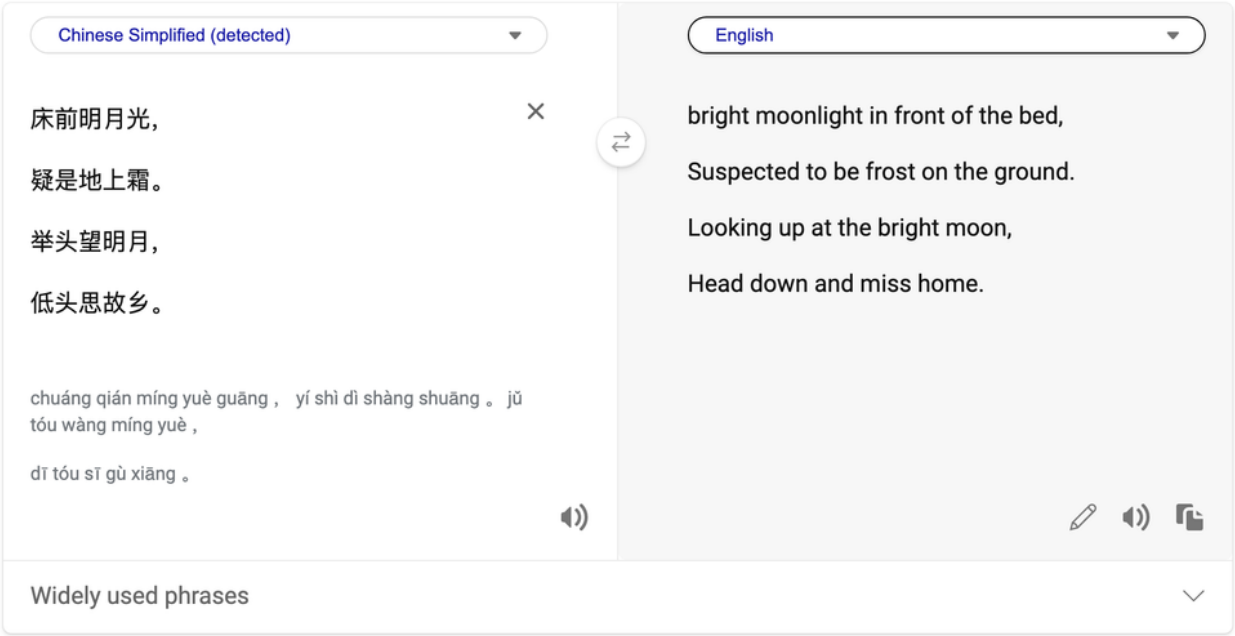
Son: They flew a lot of us to China to work with the team over there to learn the tech stack in Alibaba as well as to recode everything using the same business logic, but in Java, Spring Java of Alibaba.

Adam: Alibaba does things their own way. They have their own JVM. They have forks of Git. They have forks of MySQL. So there was a lot to learn, but Son's team was a SRE team. They didn't own any code.

Son: For 911, we merged with a team called GOC, Global Operations Center, in Alibaba. Turns out GOC is the overseeing eyes of the entire Ali Group. It was essentially a thing of SRE team in Google is the

centralized SRE team in Alibaba who measure all the monitoring system and handle all the escalation from different business unit and routed back to tech side.

Langauge Barrier



Microsoft Translate Became Son's Favorite App

Adam: So now, Son and his team are SREs for Alibaba Group, but there's a problem. The internal software used is all in Chinese, which Son can neither read nor speak. So his new project involves a lot of translation.

Son: So all the UI, all the system was in Chinese. All the documentation was in Chinese. We were over there trying to build new features and it was quite fun. It was quite an experience. I get to spend one month in Hangzhou, in Alibaba campus and boy, it was quite impressive.

Adam: So is there a language barrier?

Son: Oh, yeah, definitely. They put us in a hotel and we don't speak the language, so we couldn't order foods. We were ordering stuff by pointing fingers. Turns out you cannot get Google Translate in China by the way, so putting up your phone is not even an option. Later on, I learned that, oh, Microsoft Translate actually work, so I need to download Microsoft Translate instead, but it's still quite difficult to communicate in China and that's just living. Working is even more difficult. Everything need to go through translation, all the documentation. And sometime, the translation would be wrong, and then over time, you need to learn the dialect of the translation. For example, they'd like to translate a use case into something called scenes, like scenes in the movies. Every time you want to talk about use cases, they would translate it into scene. I was like, "What the hell is a scenes?" Turns out later on, I was like, "Oh, they meant

scenario." So over time, you just learn, pick up little, little things and everything start making a lot of more sense. But yeah, everything was very difficult.

Adam: Yeah. But how did it work to code if you were working on this ticketing system, but the other guys didn't speak English or Vietnamese?

Son: Some of them do speak English, but not at the level where you can communicate technical with them. We have to let the code speak for itself. Now, fortunately enough, I was blessed with a very talented teammate at the time and he was able to navigate through the entire full stack ecosystem from ReactJS down to Spring Java and he was impressively reteaching me a lot of that. But the app itself was quite simple. The thing about ticketing system is your CRUD and then you can attach several metadata. You have a state machine to handle different statuses. The most difficult part there is integration. Alibaba is a huge ecosystem. You have a lot of thing to integrate.

For example, you don't have Slack over there. You have DingTalk. DingTalk is a chat system of Alibaba. They also have what they call it, a change management system so that all the changes in Alibaba, all of the deployment, all the network changes can be automatically revert by one click. The moment when you deploy and some things go wrong that deviate from the metric baseline that the AI engine put out, then you have an option to click one button and revert all that change immediately. From database schema to deployment, to migration, everything, network changes, one click of a button. But yeah, it was a lot of fun.

AliPay Problems

Adam: The team Son merged with was also ops for Alipay, which became its own problem.

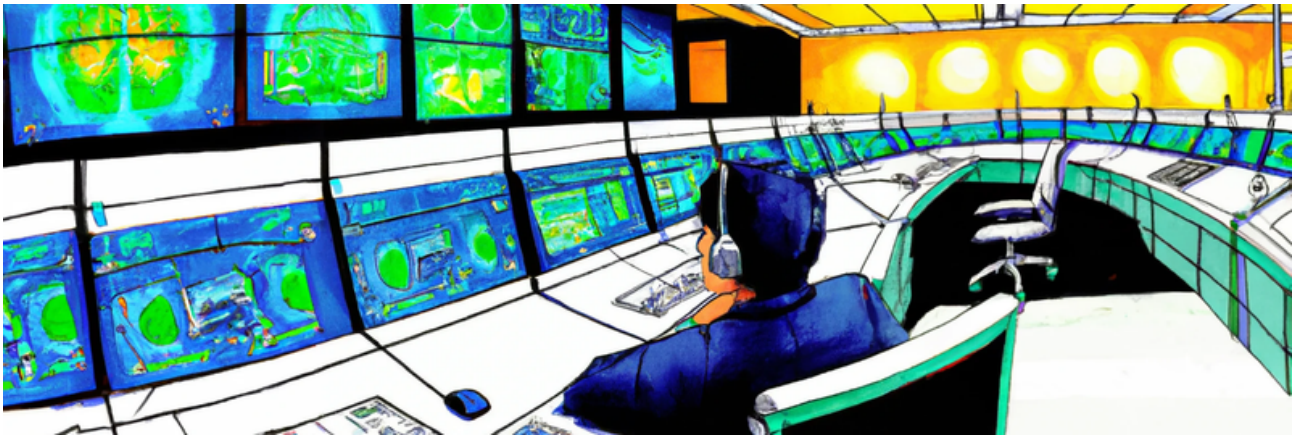
Son: In China, if you live in a megacity, people don't accept cash. When we go over there, we have a really hard time paying for food because people would only accept either WePay or Alipay. We couldn't go out and buy bubble tea because people will like, "Oh, that cash? I don't want to bother having to go to the bank and put this into the machine. That is just too much work. Alipay do." We were having to beg our teammates so that they would exchange cash that we got from Vietnam, and these are Chinese cash, it's not Vietnamese cash, we were begging them, "Hey, could you take this to the bank and then just transfer some of your Alipay credit?"

Adam: So yeah, using Alipay was a challenge, but a bigger challenge was just the scale that the Alibaba group operated at.

Son: Three rooms away from my room is a network operation center. That's where you have the big TVs with all the graphic. A Chinese team was sitting in there watching all the alerts, watching all the graph. It was very cool that you can see the all of throughput of all the platform, all the traffic, all the networks, and boy, is it massive. We were very proud of the traffic that we have in Southeast Asia. It's not even a drop in the bucket compared to China. Turns out when you have billions of people speaking the same language watching the same live stream, having the same culture, turns out that create a huge traffic spike.

Imagine you put a laptop on sales for $100 and you have 10 stock. Often in time, when you build an e-commerce system that would be one row in your SKU database, the stock keeping unit, and then that one row when somebody order it, it will change the status to sold. What happen if 10 billion people were trying to place an order on a single row in database? How do you design a system for that?

China Scale



Son's sister team worked in Alibaba's Network Operations Center. That scale of traffic was massive.

Adam: It's an interesting problem. How much is reliability worth at scale and how much does it cost? How do you build for 11/11 day when you have a billion users all at once?

Son: Things was massively impressive in China, the way they have to re-architect things in order to accommodate for the lot that they were having. I just don't see the same thing in Europe. I don't see the same thing in North America. Even the population in North America, like US and Canada combined, it's not even half the population of China. The lot we are talking about here is massive. The scale is completely massive and they don't have access to Google. They don't read Stack Overflow. They have their own stuff. When I was over there, I had to learn all of my own stuff over there and I was like, oh, my God. They just went completely different route. We were very fascinated with Golang and Kubernetes. This guys were Java, Java, Java, Java, Spring Java, Spring Java. They're just completely innovating without us knowing in Western world. Nobody was discussing how China was building stuff.

Turns out they fork their own MySQL. They were forking their own JVM. They fork their own Nginx. They fork Git so that it would use their internal object storage platform instead of on disc. So a Git hosting everywhere in GitHub, GitLab require you to have a disc, a physical hard drive, but this guy was forking it, inserting libgit2 into it and then let libgit2 talking to their internal implementation of S3. They were able to scale their DevOp platform to a cloud native scale before anybody else can. Even today, GitHub and GitLab still struggling decoupling storage and compute and this guy already have that. Similarly, they do the same thing with MySQL. They build on top of the knowledge they learn from Facebook about forking the storage engine underneath MySQL to RocksDB and then this guy were running into problem where RocksDB is an LSM-tree database storage engine.

The problem with that is that the compaction of these different layer of logs was getting quite slow and consume a lot of CBU. So guess what they did? They offload that computation to an FPGA. These guys were running custom hardware, custom chips so that their database can handle the lot better and that's just massively impressive. That's just not something that you can find in any tech company. Your custom chips like, what the hell?

996 in China

At Alibaba, working long hours is expected and Son witnessed many people napping at the office in cots or tents.

Adam: Another surprising thing at Alibaba was just how much everybody worked.

Son: 996! When I went over there, everybody told me about 996. You work from 9:00 AM to 9:00 PM six days a week, so Monday to Saturday and that's pretty much the life. Over there inside the office it's very common for you to see beds, even camping tents. Yeah, no, seriously. In Alibaba, in the hallway, you would be able to find several camping tents, gender-separated so that folks can go to sleep, not at night, but middle of the day after lunch. Sleeping in Asia is quite common, but especially during campaign night, like Double 11th. People stay over at the campus all the time.

Adam: 996 works out to 72 hours per week, but don't worry, nap time's included. At first, Son didn't like or understand this at all.

Son: But later on, I start realizing why it is the way it is. If you are a tech company and you tell people to work 9:00 to 9:00 six days a week, that's not sustainable. Nobody going to work for you. So be able to tell people to work from 9:00 to 9:00, six days a week and people still wanting to line up to work for you, there must be something special there. And turns out later on when I start talking to my Chinese teammates over there, they start telling me stories, stories that make quite a lot of sense. Think about it like this. What do you do after you gain a stable job? You get married. Before you get married in China, my friend was telling me this, you need to have at least a house and a car. If you go work for Alibaba, because they also co-own a lot of Alipay, which is a financial company that people were using instead of buying in cash, they also do loans. They can loan to you your mortgage at a very low interest rate.

The more senior you are in Alibaba, the better interest rate you get. Similar to car. You can loan from the company to buy a car and that the more senior you are, the better interest rate you get. That's not the only thing. Turns out if you work for a big company, your credit score is very high. So credit card provider would actually try to line up outside of the company so that when you left, they can hand you flyers. Here are free money, please loan from us. So that's the first thing. The second thing is the lifestyle on campus. Everybody was eating in the cafeteria. They have nine different buildings and four different cafeteria in the nine buildings. The food was being mass produced so they were able to produce that very cheap, so they were able to provide these food to the employees very cheap.

Adam: Now I haven't worked at a big tech campus, but some of this is starting to sound not that dissimilar from a Meta campus or a Google campus. Son took a lot of photos while he was there and if you want to see some of them, I've shared them on Twitter, tents to sleep in, massage stations, places to eat, and just the scale of things, but yeah, people weren't heads down working all the time and the company had lots of perks to offer them.

Son: Work day end at 6:00 officially, but the food between 6:00 and 7:00 PM would be sold at discount. So employees are incentivized to stay on campus after 6:00. Then at 9:00, they could sell you free snack. So single employees who have nothing to do at home can stay on campus until 9:00 to get those free snack. And taxi, because Alibaba was owning that taxi company, if you go home, now you get free taxi. Oh, my god. Now you start seeing all the integrations start coming into play. Hey, this guy actually provide a lot of incentive for people to work 9:00 to 9:00. Then the next thing you know, if you need to go out and buy anything, hey, Tmall and Taobao would deliver that to the doorstep of your building. You get fastest shipment on campus compared to going home.

So you would route all of your e-commerce purchases to the company address. Just tell what building is it and then go down and get it. So it's more convenient to live on campus compared to going home. People was getting all the perks everywhere that affecting their life. They're willing to work for it.

Ladder Climbing and KPIs

Adam: All these perks means a lot of people want to work there, and then the higher you climb the corporate ladder, the more perks they are, which leads to the fact that internal competition can get pretty intense.

Son: They're willing to compete for the jobs and that's why Alibaba had a very cutthroat culture so that they can digest out low performers every years. I think 30% of the low performers are digest out. Everybody was competing for the top because turns out, one more perks, which is your final bonus, your yearly annual bonus can be up to 100% of your salary.

Adam: Oh, wow. That's huge.

Son: So people was competing to be the top performers. This guy know what they were doing.

Adam: How do you manage a giant company of software developers who all want to work really hard and all really want to advance their careers? Well, maybe you already know the answer.

Son: The way Alibaba manage is very top-down, very KPI-heavy. Every business unit are given a certain KPI. The leader of those business unit would assign KPI to each of the sub-department and each of the sub-department would assign KPI down to their employees. They have metrics to meet. This guy were very metric-driven.

Adam: This KPI or OKR system, much like any top-down system thrown at motivated people, leads to some unintended consequences. For instance, porting Lazada to the internal platform was one thing, but actually running the company, maybe that was a less exciting project.

Son: This is hearsay, so I'm not saying this is a source of truth, but there were rumors to be like they think Lazada to be a not profitable place for them to meet the KPI, so nobody wanted to be a CEO of a failing position. Lazada actually changed CEO quite a lot because nobody was meeting their KPI. The culture there is quite cutthroat and people were competing for nice position so that they can be set up for success.

Adam: There's a way it sounds very foreign. There's another way that it sounds like Google.

Son: Well, I cannot tell you because I have never worked for Google, but what I can tell you is that they move very fast and I don't think anybody has been moving as fast. Especially if you work in a Western country, in European, or in North America, you are not moving fast. This guy move a lot faster.

Hello,
I make CoRecursive because I love it when someone shares the details behind some project, some bug, or some incident with me.

No other podcast was telling stories quite like I wanted to hear.

Right now this is all done by just me and I love doing it, but it's also exhausting.

Recommending the show to others and contributing to this patreon are the biggest things you can do to help out.

Whatever you can do to help, I truly appreciate it!

Thanks! Adam Gordon Bell

Support The Podcast

Singapore

Son went to work in Singapore, who vibrant airport affected how Son thinks about planning.

Adam: Okay. So Alibaba and China move fast and that can be exciting, but for Son, it had downsides.

Son: Alibaba obviously providing a lot, but I was interested in learning what else is there and I was a little bit burned-out with all of the Google Translate of Chinese documentation. I was looking for other opportunities and my boss was like, "Hey, I'm going to the exact same company but in Singapore."

Adam: So Son stayed working for Lazada, but moved to Singapore where his time on site made him the expert for his team.

Son: Then I get to teach my teammates about Alibaba tech. They were using Python at the time, a lot of Python, Django, Flask self-hosted on Ali Cloud and I was like, "No, guys. Here's Ali Java. Here's all the goodness. Hey." Imagine Spring Java, but everything come with battery-equipped. You want a database? Here's a database. You want a message queue? Here's a message queue. You don't need to worry about infrastructure. Everything came out of the box, and that's was actually a downside because I start to realize that the world outside was moving quite fast. Even though I was quite an expert in internal technology in Alibaba, I know nothing about the world outside.

Adam: Son wanted to keep growing and learning and not pigeonhole himself into one company, into one tech stack, so he set a deadline for himself.

Son: So I set up a plan that after I arrive in Singapore for a year, I would need to learn new technology and explore the job market. At the time, I was actually interviewing with Google in Singapore. I attend several of their tech talk, get to talk to the G Pay team over there in Singapore. They were doing a lot of interesting things.

Adam: Google famously has a super long hiring process and in the meantime, Son found an opportunity to do something completely different.

Son: At the time, I was getting to know my girlfriend at the time and now my wife. I was like, okay, I need to settle down someplace and good luck buying a house in Singapore. Singapore was way too expensive. Your best luck is being able to rent a nice apartment and that's pretty much it. You would not even get a car. It would cost you millions to get a cars and that's insane.

Amsterdam

Son moved to Amsterdam to work for booking.com

Adam: Son got an offer from Booking.com. Booking.com is an online travel agency based in the Netherlands in Amsterdam, and in some ways, Amsterdam is the opposite of Singapore.

Son: I took that opportunity and said goodbye to Singapore even though I love Singapore so much. It's such a nice country, wonderful, strongly recommend.

Adam: Where's the food better?

Son: Oh, Singapore, definitely. Netherland is like the worst. Oh, my god. No, seriously. Even Dutch hate the food in the Netherland. So I left Singapore for the Netherland August 2019, and then me and my wife officially get married on the cusp of COVID.

Work Life Balance

Son: I remember my product manager on the first day I came to work. He sit me down with my team lead. My team lead was actually joining with me. We did the orientation together. My product manager sit both of us down and explain, "Hey, guys. Booking paid a lot of money to relocate you guys here. We want you to success here. That means you need to prioritize your life first. You need to be able to find a house to rent. You need to be able to do all your paperwork. So in the next upcoming month, if you need to take time out, just go and then come back and tell us later," and that is such a huge opener because that's just a mental shift, completely 180 compared to Alibaba.

Because Alibaba, everything was work, work, work. Your life is in the company. Your metric is important. Booking was not like that. Booking was like, "No. We respect you as a human. We know that you have a family. We know that you have needs. Take care of them and then come back to us. Work your best," and that's just changed everything. In Booking, I have teammates who are blind. I have teammates who are gay, lesbian, disabled. We were all able to hang out together. No problem. There's no discrimination. It's just a huge, open, and welcome culture. Very diverse pool of folks over there. I just love it so much. We still have massive traffic. We still have problem at scale that we have to take care of and yeah, it was very

fun trying to find work-life balance in midst of all that, in midst of COVID, in midst of relocation, and in midst of new marriage as well.

Reflection

Son's advice is that developers should travel more.

Adam: So for now, Son has found his place in the world. He and his wife are still in the Netherlands. He found his home and a place where the work culture works best for who he is, but he only knows that because of the experiences he had. If he had started at Booking first thing, maybe he wouldn't have realized what he had because he'd have no comparison. That fact shaped the advice that he offers to others:

Son: Life is diverse. Before I get to work in China, it was a black box to me. It was not occurred to me that there was a payment system that's so advanced that people would refuse using cash. Yeah, I know credit card exists, but that doesn't mean when I hand you cash, you will refuse to receive it. Just at the same time, going to Southeast Asia, seeing various different startup trying to get themself a place on the map was hugely game-changing because it enabled me to think in term of business, in term of thinking, how would I operate this if I were the CEO or CTO? I also learned how to party. In Waterloo, they drink bubble tea. You can get way better party going to Toronto than trying to do it in Waterloo. Going back to Vietnam, every one week or two weeks is a drinking party.

A lot of beers we consume, a lot of hot pot. The food is wonderful over there. When I go to Singapore, I learn what planning can bring. You can just enter the city and you can see. The first thing you see in Singapore is the best, most beautiful looking airport in the world. You can see like, "Oh, my God. How much would it take for my government to build this airport?" Infinity money would not change this because this is not a problem with money. This is a problem with planning. Then Netherland, they taught me to be open-minded. They taught me about work-life balance and all that is just come back to be travel more.

This is not an advertisement for Booking.com, but travel more and learn more from it and keep an open mind, because yeah, maybe Western media can tell you a lot of shit about China, about how evil things can be, but try to go over there and try to see how people live. Being able to see thousands of people eating

lunch all at once surrounding you or looking like you, because I'm Asian, make you humble. It does have a humble feeling entering that cafeteria and see hundred of people lining up for the same dish. It make you feel insignificant. **It's a massive world out there. Travel more.**

Outro

Adam: That was the show!

Thank you so much to Son! You can find him on Twitter. I'll put a link on the episode page.

In an upcoming bonus episode for podcast supporters, Son is going to share his views on the future of Chinese software development and a little bit about the Chinese expansion via the Belt and Road Initiative.

It's super interesting stuff!

A lot of the innovations that Alibaba is doing are out there on GitHub and they're open sourced and they're for the taking, but outside of certain system researcher circles, it feels like nobody's paying attention. That's going to change. So support the podcast and on Patreon, you can get access to that episode that I'll release in the future.

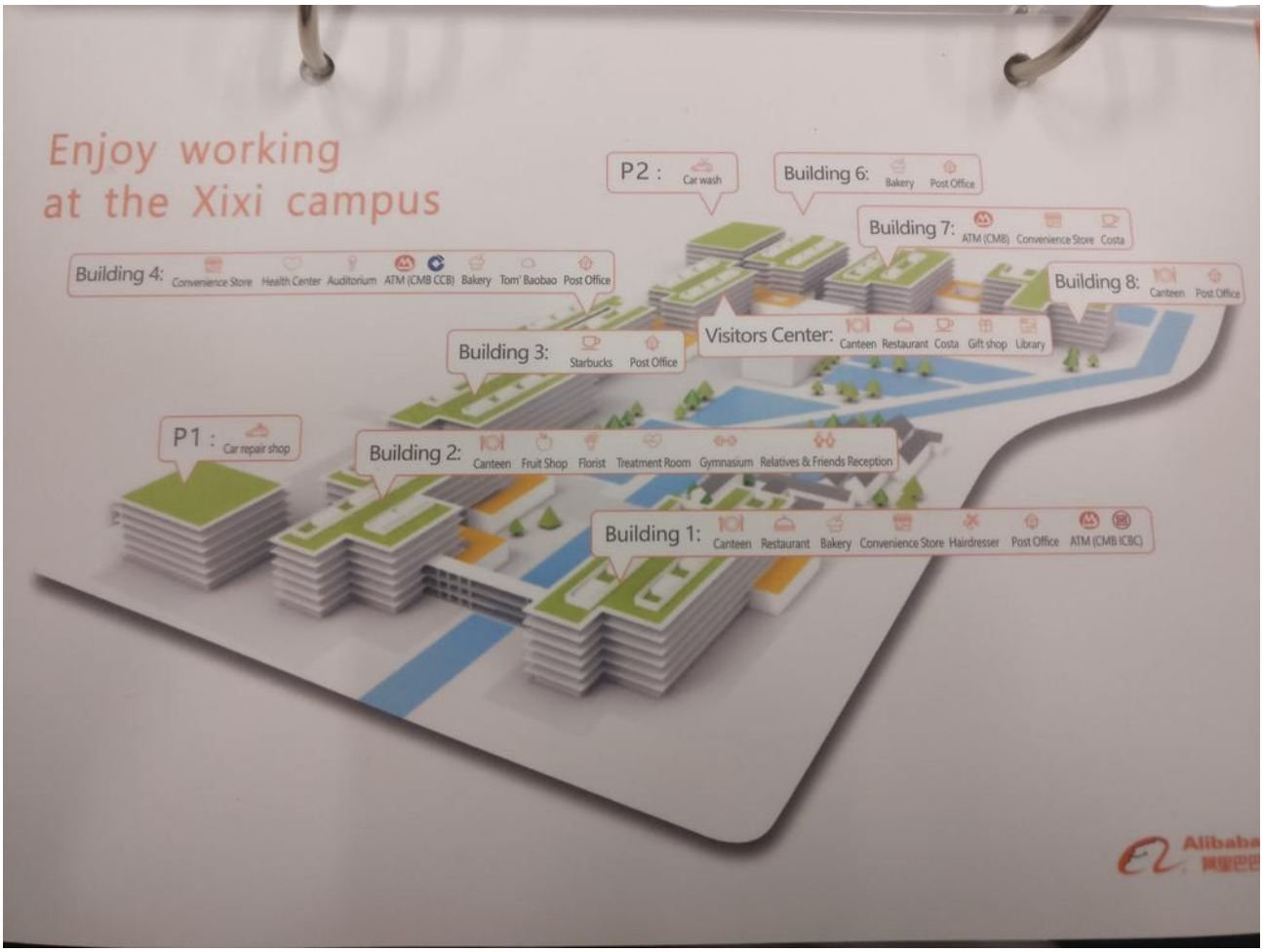
Also, Son's career keeps evolving, so you'll hear a little bit about his latest career pivot, which is how I ended up meeting him. Also, if you like to hear a little bit of the fun backstory behind the episodes, then this'll be a great opportunity to hear that.

Until next time, thank you so much for listening.

Bonus Content From Son

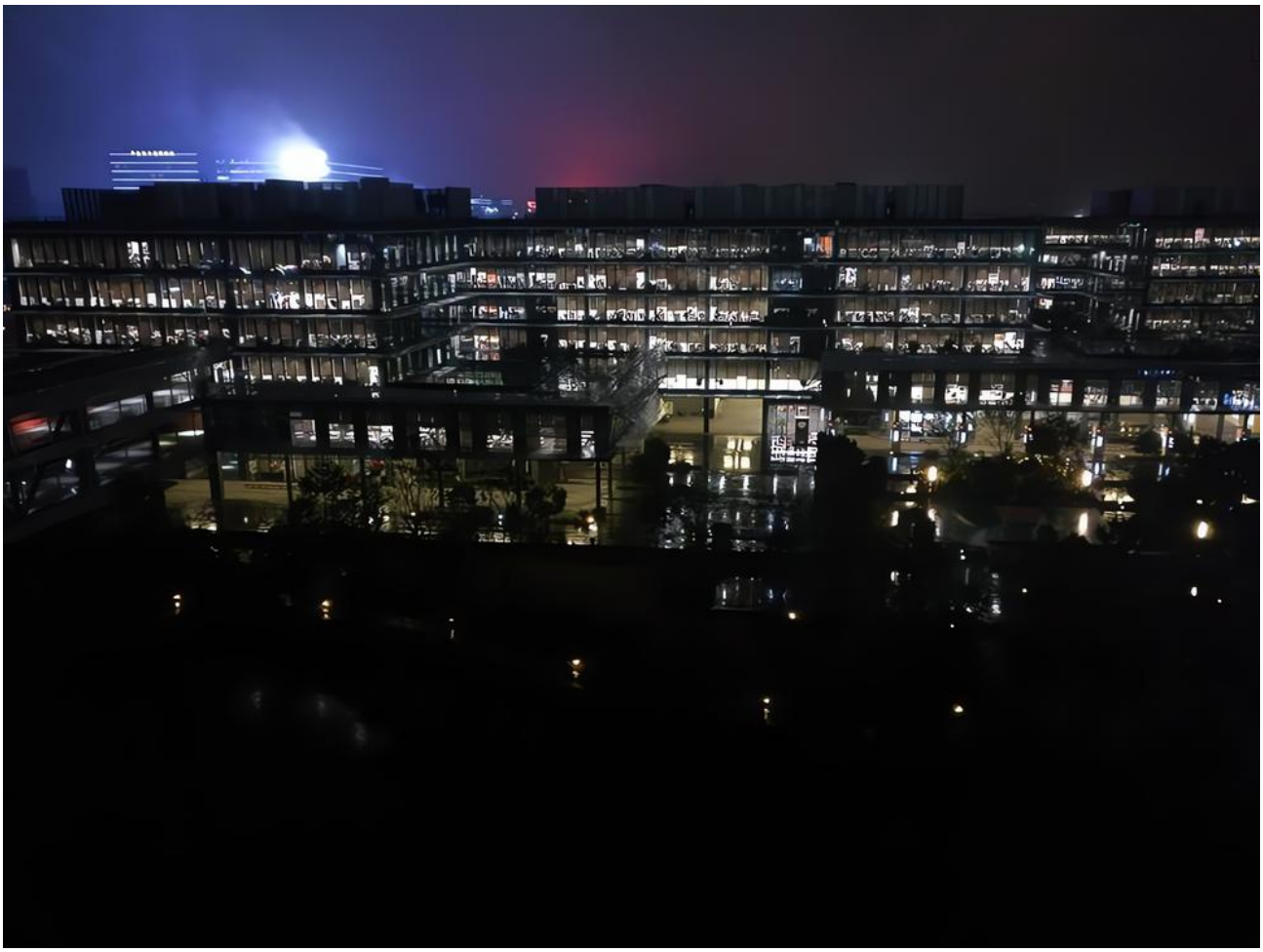
Son: I think just words could be pretty dry, so here are some visual aid from my trip to Alibaba Campus in 2018.

Son: Here is the main campus map. U shaped with the main residence of Jackma being in the middle of it (the grey part)



Son: Here is the campus by day:

Son: Here is the campus at 7-9PM, 9 buildings fully lit on all floors:



Son: Here are the tents in the hall way, gender separated for folks to sleep in whenever.

Son: For satelite campuses, offices would have bed like this instead of the tents:

Son: The entrance to each building is equipped with badge scanner, facial scanner was newly added in summer 2018:

Son: On campaign days, there would be maids handing out swags to engineers:

Son: or free massages from professionals

Hello,

I make CoRecursive because I love it when someone shares the details behind some project, some bug, or some incident with me.

No other podcast was telling stories quite like I wanted to hear.

Right now this is all done by just me and I love doing it, but it's also exhausting.

Recommending the show to others and contributing to this patreon are the biggest things you can do to help out.

Whatever you can do to help, I truly appreciate it!

Thanks! Adam Gordon Bell

Support The Podcast